主要内容

- 密码技术应用（续）
  - 对称密钥加密算法
  - 公钥密码体系概述
    - 非对称加密算法
    - Diffie-Hellman密钥交换算法

加密方法分类

三种分类方法:

- 从明文到密文的变换
  - 替换(substitution)
  - 置换(transposition)
  - ……
- 密钥的数目
  - 对称、单钥加密法
  - 公钥、双钥加密
- 明文的处理方式
  - 分组加密（块加密算法）
  - 流方式加密

密码算法

- 古典密码算法
  - 替换技术(Substitution)
  - 置换技术(Transposition)
- 现代密码算法
  - DES
  - 3DES
  - AES
  - 其他密码算法

替换和置换

- 替换技术(Substitution)
  - 在加密时将明文中的每个或每组字符由另一个或另一组字符替换，原字符被隐藏起来，即形成密文
- 置换技术(Transposition)
  - 就是在加密时只对明文字母（字符、符号）重新排序，每个字母位置变化了，但没被隐藏起来。置换密码是一种打乱原文顺序的加密方法

替换技术(substitution)

◆凯撒（Caesar）密码：最早的替换密码

- ❖ 每个字母用它之后的第3个字母来代换（密钥k=3）
- ❖ 弱点：26个字母（密钥）

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

数值映射：

a b c d e f g h i j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

加/解密算法 Caesar cipher：

$c = E(k, p) = (p + k) \bmod (26)$

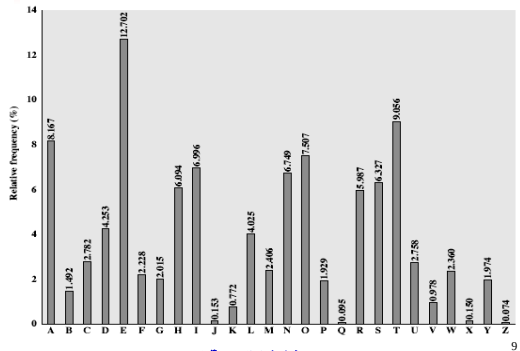
$p = D(k, c) = (c - k) \bmod (26)$

替换技术(实例)

Plain Text and Cipher Text

Plain text message	Corresponding cipher text message
Hi Amit, Hope you are doing fine. How about meeting at the train station this Friday at 5 pm? Please let me know if it is ok with you. Regards. Atul	**Kl Dplw,** **Krsh brx duh grlqj ilqh. Krz derxw phhwlqj dw wkh wudlq vwdwlrq wklv Iulgdb dw 5 sp? Sohdvh ohw ph nqrz li lw lv rn zlwk brx.** **Uhjdugv.** **Dwxo**

英文字母的频率



置换技术(transposition)

◆栅栏加密技术

- ❖用对角线方式写明文，然后按行重新排序

C m h m t m r o

o e o e o o r w

Cmhmtmrooeoeoorw

◆简单分栏

- ❖将明文消息一行一行地写入预定长度的矩形中
- ❖一列一列读，随机顺序
- ❖密文消息

多轮简单分栏

明文消息为 Come home tomorrow。

（1）假设矩形为 6 列，则可以将明文消息一行一行地写入其中如下：

第 1 列	第 2 列	第 3 列	第 4 列	第 5 列	第 6 列
C	o	m	e	h	o
m	e	t	o	m	o
r	r	o	w		

（2）下面要指定随机列顺序，如 4, 6, 1, 2, 5, 3,然后按这个顺序一列一列读消息。

（3）得到密文 eowoocmroerhmmto，这是第一轮

（4）下面再次执行 1~3 步，第 1 轮后的密文表格如下：

第 1 列	第 2 列	第 3 列	第 4 列	第 5 列	第 6 列
e	o	w	o	o	c
m	r	o	e	r	h
m	m	t	o		

（5）假设使用相同的随机列顺序，即 4, 6, 1, 2, 5, 3，然后按这个顺序一列一列读消息。

（6）得到密文 ocochemmormorwot。

（7）可以进行多次迭代，也可以到此为止。

古典密码算法特点

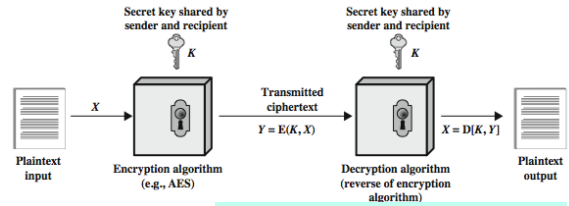
- 特点
  - DES之前
  - 要求的计算强度小
  - 以字母表为主要加密对象
  - 替换和置换技术
  - 数据安全基于算法的保密
- 密码分析方法
  - 基于明文的可读性
  - 字母和字母组合的频率特性

现代密码算法

- 对称加密算法
  - 数据加密标准 DES（Data Encryption Standard）
  - 高级加密标准AES（ Advanced Encryption Standard ），密钥长度：128，196，256位
  - 三重DES (3DES)

对称密钥加密模型

Symmetric Cipher Model



$Y = E(K, X)$ 加密
$X = D(K, Y)$ 解密

K=密钥 （Secret Key）
加密密钥和解密密钥相同
也称为：单一密钥Single-key or 私有密钥加密private key encryption.

对称密钥加密算法

◆加密、解密算法是公开的

◆安全性完全取决于密钥*K*的安全性

保证密钥安全性的两种机制：

◆一次一密，密钥之间没有任何相关性

◆在公开加密、解密算法的条件下，即使获取多对明文/密文对也很难推导出密钥K

密码分析 Cryptanalysis

◆密码分析攻击 Cryptanalytic Attacks

- ❖依赖于算法性质
- ❖明文一般特征
- ❖明文-密文对样本

◆利用算法特征推导出特定的明文或使用的密钥

- ❖ 灾难性

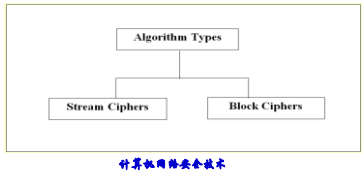
◆暴力攻击Brute-Force Attack

- ❖蛮力攻击，穷举攻击
- ❖对一条密文尝试所有可能的密钥，直到获得明文
- ❖平均至少要尝试所有可能密钥的一半

算法类型

◆序列(流)密码与分组密码

- ❖序列密码(stream cipher)也叫流密码，是对单个明文位(Bit-by-bit)变换的操作
- ❖分组密码(block cipher)是对一个大的明文块(Block-by-block)进行固定变换的操作
  - ➢典型分组大小：64位或128位



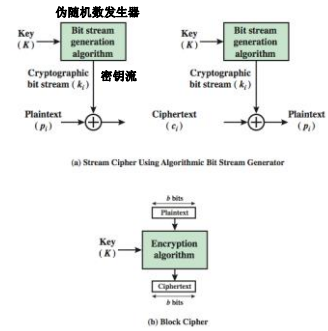
Stream vs. Block Ciphers

◆流密码: Bits and bytes are processed as they arrive

- ❖ 例: RC4

◆分组密码（块密码）: Messages are broken into blocks of 64-bit, 512-bit, ...

- ❖ 使用最广泛的对称加密算法
- ❖ 例: DES, AES



分组密码算法设计指导原则

- 混淆Confusion
  - 强调密钥的作用
  - 增加密钥与密文之间关系的复杂性
  - 流加密和分组（块）加密
- 发散Diffusion
  - 小扰动的影响波及到全局
  - 密文没有统计特征，明文一位影响密文的多位，增加密文与明文之间关系的复杂性
  - 分组（块）加密

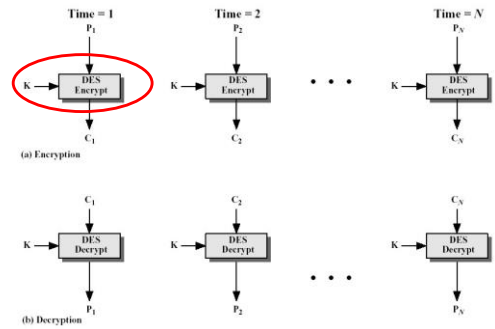
分组密码算法模式

- 电子簿模式(ECB，electronic codebook mode)
- 密码块链接(CBC，cipher block chaining)
- 密码反馈方式(CFB，cipher feedback)
- 输出反馈方式(OFB，output feedback)
- 计数器（CTR， Counter ）

电子簿模式ECB

- 采用相同的密钥分别对64bit明文组进行加密
- 优点
  - 简单；有利于并行计算；误差不会被传送
- 缺点：不能隐藏明文的模式，可能对明文进行主动攻击
  - 相同明文⇨相同密文
    - 同样信息多次出现造成泄漏
    - 信息块可被替换
    - 信息块可被重排
  - 密文块损坏⇨仅对应明文块损坏
- 适合于传输短信息

电子簿模式ECB

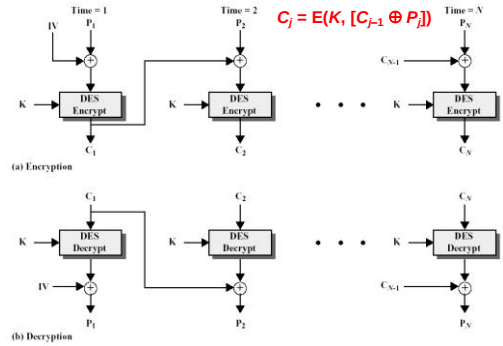


密码块链接CBC

- 加密算法的输入是上一个64bit密文组和下一个64bit明文组进行“**异或**”运算
- 需要共同的初始化向量IV
  - 相同明文⇨不同密文
  - 初始化向量IV可以用来改变第一块
- 优点
  - 不容易被主动攻击
  - 安全性好于ECB
  - 适合传输长度长的报文，是SSL、IPSec的标准。
- 缺点
  - 不利于并行计算
  - 误差传递：一密文块损坏⇨两明文块损坏

密码块链接CBC

$C_j = E(K, [C_{j-1} \oplus P_j])$



分组密码的优缺点

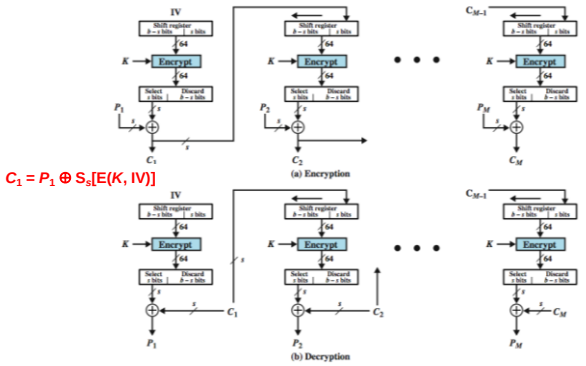
- 分组密码主要有两个优点
  - **易于标准化**
  - **易于实现同步**
- 局限性
  - **分组密码不便于隐藏明文的数据模式**
  - **对于重放、插入、删除等攻击方式的抵御能力低**

通过采用流密码的设计思想，在加密过程中采用合理的记忆组件，能够消除这些局限性

密码反馈方式CFB

- 上一个分组密文作为产生一个伪随机数输出的加密算法的输入，该输出与明文“异或”，作为下一个分组的输入
- 优点
  - 隐藏了明文模式
  - 分组密码转化为**流密码**
  - 可以及时加密传送小分组的数据
- 缺点
  - 需要共同的移位寄存器初始值IV
  - 对于不同的消息，IV必须唯一
  - 一个单元损坏影响多个单元：(W+j-1)/j
    W为分组加密块大小，j为流单元位数
  - 不利于并行计算

密码反馈方式CFB



$C_1 = P_1 \oplus S_s[E(K, IV)]$

输出反馈方式OFB

- 与CFB基本相同，只是加密算法的输入是上一次加密算法的输出
- OFB:分组密码⇨流密码
- 需要共同的移位寄存器初始值IV
- 一个单元损坏只影响对应单元

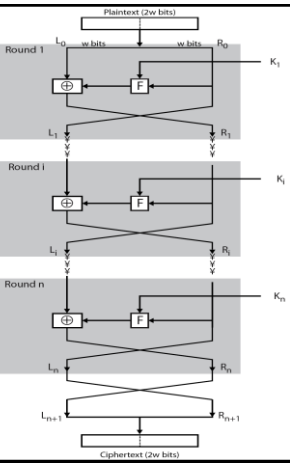
计数器Counter (CTR)

- 计数模式（CTR模式）加密是对一系列输入数据块(称为计数)进行加密，产生一系列的输出块，输出块与明文异或得到密文。
- 对于最后的数据块，可能是长u位的局部数据块，这u位就将用于异或操作，而剩下的b-u位将被丢弃（b表示块的长度）
- CTR解密类似。这一系列的计数必须互不相同的。
- CTR 模式被广泛用于 ATM 网络安全和 IPSec应用中

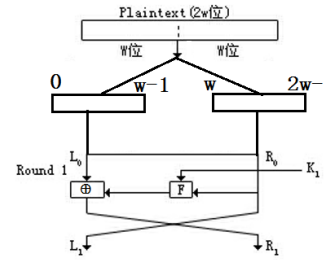
Feistel结构的分组加密算法

- 基本思想：用简单算法的乘积来近似表达大尺寸的替换变换
- 多个简单算法的结合得到的加密算法比任何一个部分算法都要强
- 交替使用替换置换和排列(permutation)
- 混淆(confusion)和发散(diffusion)概念的应用

Feistel 结构图



Feistel结构定义

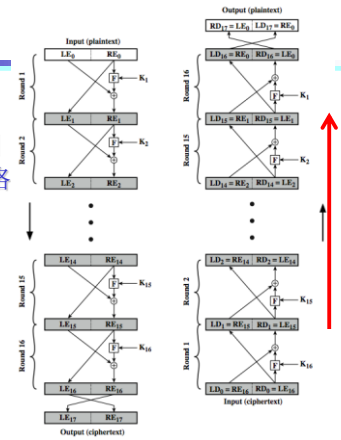


- 先将2w位的明文分为左半部分（前w位L0）和右半部分（后w位R0）
- 将输入的右侧R0，直接输出到输出的左侧为密文的左半部分L1。
- 将输入的右侧R0与子密钥K1进行F函数操作（K1和R0作为自变量）得到运算结果，即
  Output_1 = F（K1,R0）。
- 将经过F函数运算的结果Output_1与L0进行异或操作，得到结果作为密文的右半部分R1。
- 第一轮Round1的密文作为第二轮Round2的明文进行相同步骤的加密操作，循环多轮操作。

Feistel结构定义

- 加密: $L_i = R_{i-1}$;
  $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- 解密: $R_{i-1} = L_i$
  $L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$
  $= R_i \oplus F(L_i, K_i)$

经典Feistel加/解密网络



Feistel分组加密算法的特点

- 分组大小：越大安全性越高，但速度下降，128bit比较合理
- 密钥位数：越大安全性越高，但速度下降，128bit广泛使用
- 迭代轮数：典型16轮
- 子密钥产生算法：算法越复杂，增加密码分析的难度
- 轮函数：函数越复杂，增加密码分析的难度
- 快速软件实现：包括加密和解密算法
- 易于分析：便于掌握算法的保密强度以及扩展办法

DES算法

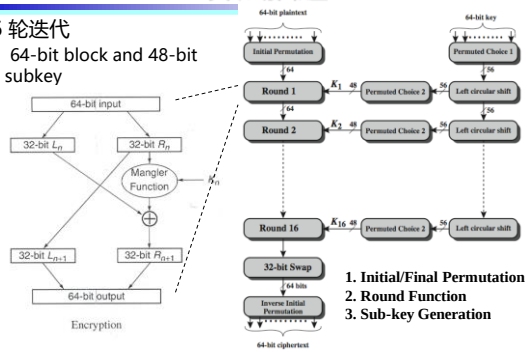
- 1977年由美国国家标准局(NBS，现为NIST)采纳
  - 使用64 bit明文块和 56 bit密钥，生成 64 bit 密文块
- 历史：
  - IBM在60年代启动了LUCIFER项目，当时的算法采用128位密钥
  - 改进算法，降低为56位密钥，IBM提交给NBS(NIST)，产生DES
- 16轮的Feistel结构密码
- 主要用于金融交易
- 安全性：DES→ 3DES（Triple DES）

DES算法描述

- DES是对称密钥加密的算法， 大致可以分成四个部分：
  - 初始置换
  - 迭代过程
  - 逆初始置换
  - 子密钥生成

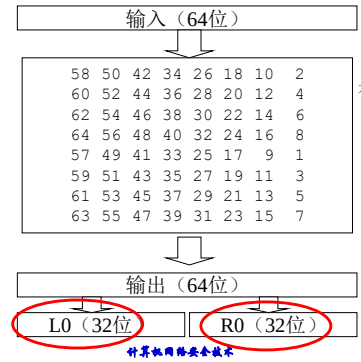
DES算法描述

- 16 轮迭代
  - 64-bit block and 48-bit subkey

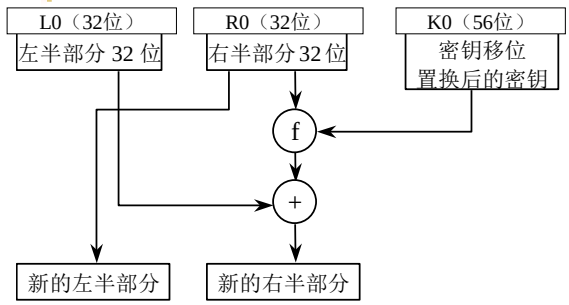


1. Initial/Final Permutation
2. Round Function
3. Sub-key Generation

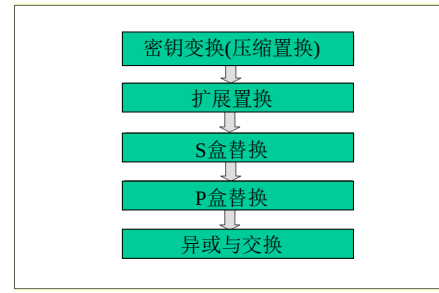
初始变换IP



乘积迭代原理示意图



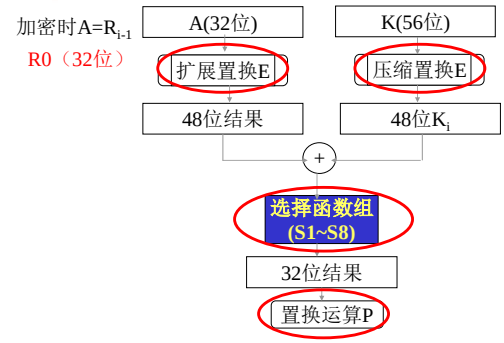
DES的一轮（f）



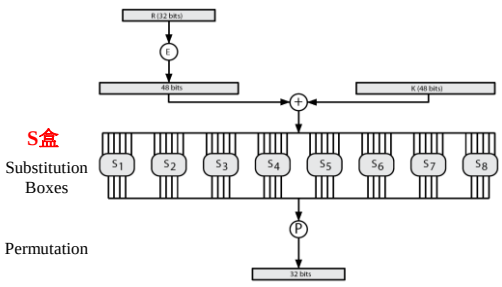
S盒运算

- 在密码函数中有8个S盒，称为8个不同的选择函数。每个S盒都是将6位（6*8）块作为输入，得到一个4位（4*8）块作为输出
- S 盒是DES 的最敏感部分，其原理至今未公开。人们担心S 盒隐藏陷门，使得只有他们才可以破译算法，但研究中并没有找到弱点

加密函数



DES 轮结构（Round Structure）



P盒置换

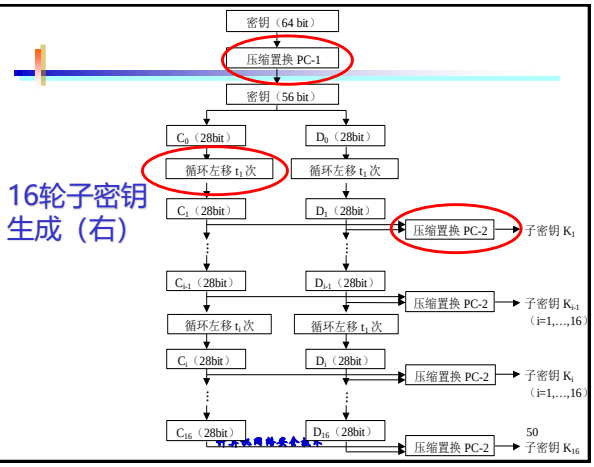
8个选择函数的输出 （32位）

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

置换P

加密函数的结果X （32位）

16轮子密钥生成（右）



DES加密和解密过程比较

解密：

由密文到明文的解密处理和由明文到密文的加密处理类似。两者使用同一组子密钥，不同的只是两者的生成次序正好相反，即解密时用到的第一个子密钥K_1是加密时最后生成的子密钥K_{16}，依此类推。

下面对比一下使用DES算法进行加密和解密的处理过程。

（1）加密过程

L_0R_0←64bit明文经IP置换

L_i←R_{i-1}(i=1,…,16)

R_i←L_{i-1} ⊕ f(Ki,Ri-1)，(i=1,…,16)

64bit密文bit←$R_{16}L_{16}$经IP^{-1}置换

（2）解密过程

$R_{16}L_{16}$←64bit密文经IP置换

R_{i-1}←L_i(i=16,…,1)

L_{i-1}←R_i ⊕ f(K_i, L_i)(i=1,…,16)

64bit密文bit←R_0L_0经IP^{-1}置换

DES的特点及应用

（1）DES算法的特点

DES算法具有算法容易实现、速度快、通用性强等优点；但也有密钥位数少、保密强度较差和密钥管理复杂等缺点。

（2）DES的主要应用

① 计算机网络通信。对计算机网络通信中的数据提供保护是DES的一项重要应用，但这些保护的数据一般只限于民用敏感信息，即不在政府确定的保密范围之内的信息。

② 电子资金传送系统。采用DES的方法加密电子资金传送系统中的信息，可准确、快速地传送数据，并可较好地解决信息安全的问题。

③ 保护用户文件。用户可自选密钥，用DES算法对重要文件加密，防止未授权用户窃密。

④ 用户识别。DES还可用于计算机用户识别系统中。

DES的安全分析

- ◆DES算法的本质
  - ❖关键在于8个S-BOX
- ◆针对DES的密码分析
  - ❖差分分析法
    - Biham和Shamir于1991年提出
    - 属于选择明文攻击
    - 基本思想：通过分析特定明文差对结果密文差的影响来获得可能性最大的密钥
    - 2^{47}对选择明文，经过2^{47}量级的计算可攻破
  - ❖线性分析法
    - Matsui和Yamagishi于1992年
    - 思想：用线性近似描述DES变换
    - 根据2^{47}已知明文，可以找到DES的密钥

DES的安全分析

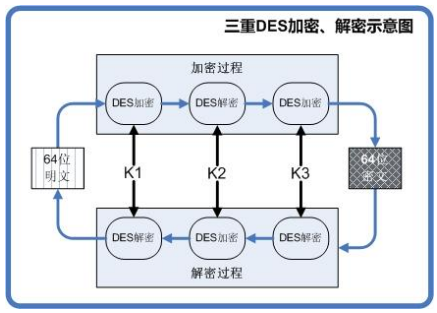
- S盒设计
- 弱密钥或半弱密钥
- DES结构的互补对称性
  - 不要使用互补密钥
- 穷举攻击
  - 56位密钥的使用，理论上，97年$100000的机器可以在6小时内用穷举法攻破DES
  - 实际攻破的例子，97年1月提出挑战，有人利用Internet的分布式计算能力，组织志愿者连接了70000多个系统在96天后攻破
  - 以后又分别用41天、56个小时和22个小时破解了用56bit DES算法加密的密文
  - 目前，多核高性能计算机，加密速度10^{13}次/秒，1小时破解56bit DES

三重DES（3DES）

- 重复基本 DES算法三次，采用两个或三个不同密钥
- 优点
  - 168-bit密钥，克服DES面临的暴力攻击
  - 底层加密算法与DES相同
- 缺点
  - 用软件实现算法速度较慢
  - 分组长度64-bit

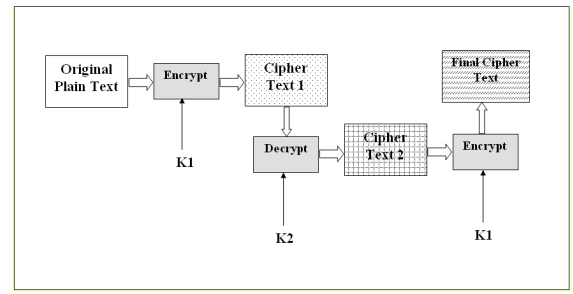
三重DES（3DES）

- 三个密钥（E-D-E）



三重DES

- 两个密钥



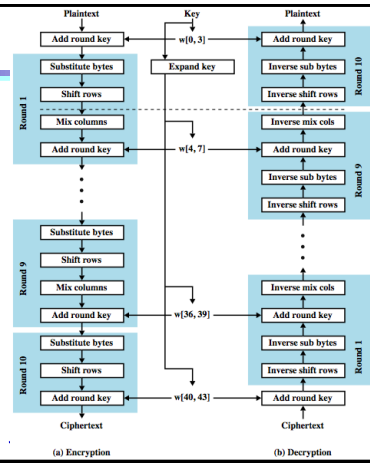
AES算法

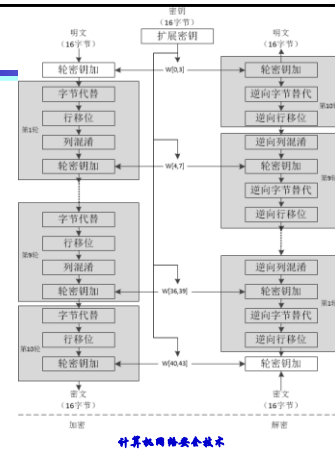
- 高级加密标准（Advanced Encryption Standard，AES）是由美国国家标准技术研究所（NIST）于1997年发起征集的数据加密标准
  - 安全强度不低于3DES，显著提高计算效率
  - 非保密的、全球免费使用的分组加密算法，并成为替代DES的数据加密标准
- NIST于2000年选择了比利时两位科学家提出的Rijndael算法作为AES的算法
- Rijndael算法具有安全、高效和灵活等优点，使它成为AES最合适的选择
- 分组大小128bit，密钥长度128，192或256bit。
- 没有使用Feistel结构

AES算法加密过程

- AES加密过程涉及到4种操作
  - 字节替代（SubBytes）
  - 行移位（ShiftRows）
  - 列混淆（MixColumns）
  - 轮密钥加（AddRoundKey）
- 解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。
- 加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。

AES





三种流行对称加密算法比较

	DES	Triple DES	AES
明文块 (bits)	64	64	128
密文块 (bits)	64	64	128
密钥 (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

穷举密钥搜索所需平均时间

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

IDEA算法

- 国际数据加密算法（International Data Encryption Algorithm，IDEA）是由瑞士的著名学者首先提出的，1990年被正式公布并在随后得到了增强。这种算法是在DES算法的基础上发展起来的，类似于三重DES
- IDEA算法设计了一系列加密轮次，每轮加密都使用从完整的加密密钥中生成的一个子密钥。每轮次中也使用压缩函数进行变换，只是不使用移位变换
- IDEA把数据分为4个子分组，每个分组16bit

IDEA算法

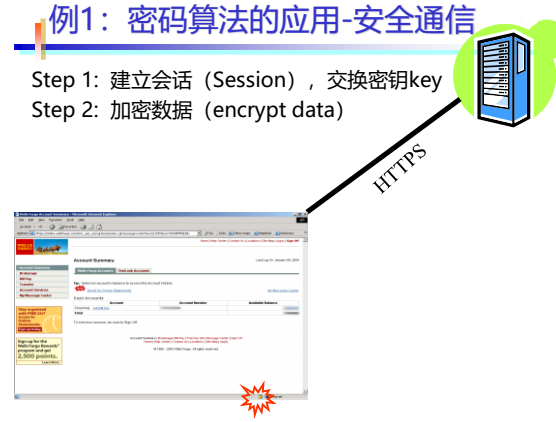
- IDEA算法可用于加密和解密。主要有三种运算：异或、模加、模乘，容易用软件和硬件来实现
- IDEA的速度：现在IDEA的软件实现同DES的速度一样块
- IDEA的密码安全分析：IDEA的密钥长度是128位，是DES的密钥长度的两倍。在穷举攻击的情况下，IDEA将需要经过2^{128}次加密才能恢复出密钥

RC5

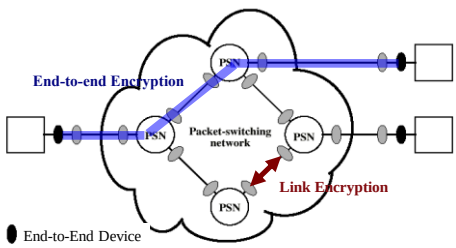
- RC5是Ron.Rivest 于1994年设计的一种新的分组算法。
  - 前身RC2、RC4分别是可变密钥长度的分组和流加密算法
  - RC5是可变密文长度、可变轮数、可变密钥长度的分组加密算法
- RC5加密算法的特点有：
  - 基本运算是微处理器上常见的初等运算
  - 字的位数作为RC5的参数
  - 安全性依赖于旋转运算和不同运算的混合
  - 存储要求低，适合在智能卡上实现
  - 轮数和密钥长度可以变化
- RC5算法由密钥扩展算法、加密算法、解密算法组成
- 应用
  - RC4应用于SSL/TLS，WEP, WPA等

例1：密码算法的应用-安全通信

Step 1: 建立会话（Session），交换密钥key
Step 2: 加密数据（encrypt data）



例2：分组交换网上的加密

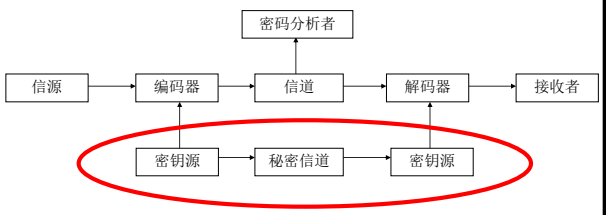


End-to-End Device
Link Encryption Device
PSN = Packet Switching Node

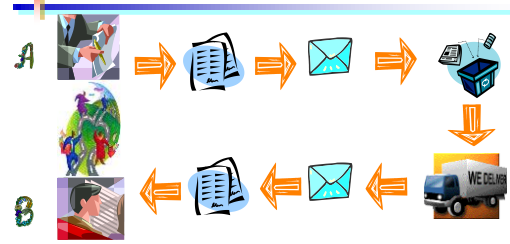
Encryption Across a Packet-Switching Network

公钥密码体系概述

密码系统的模型：密钥分发

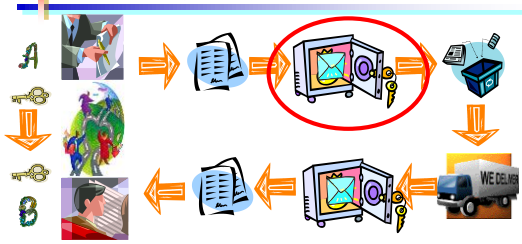


对称与非对称密钥加密



- 传递信息是否安全?
- A ⇒ P ⇒ B

对称与非对称密钥加密



- 使用钥匙是否可保证安全?
- A ⇒ B
- 密钥分发（Distribution）或交换（exchange）

Diffie-Hellman密钥交换协议 / 算法

密钥交换协议 / 算法的历史

- 1976年Diffie和Hellman发表了“密码学的新方向”
  - 奠定了公钥密码学的基础
- Diffie-Hellman密钥交换协议 / 算法
  - 使用此方法确定对称密钥交换
- 1978年，RSA算法
- 公钥技术是二十世纪最伟大的思想之一
  - 改变了密钥分发的方式
  - 可以广泛用于数字签名和身份认证服务

算法示例

1. Alice与Bob确定两个大素数n及其本原根g，这两个整数不保密， Alice与Bob可以使用不安全信道确定这两个数.
   设 n=11,g=7
2. Alice选择另一个大随机数 x，并计算A如下:
   $A=g^x \bmod n$
   设 x=3,则 $A = 7^3 \bmod 11 = 343 \bmod 11 = 2$
3. Alice将A发给Bob
   Alice将2发给Bob
4. Bob选择另一个大随机数 y，并计算B如下:
   $B=g^y \bmod n$
   设 y=6,则 $B = 7^6 \bmod 11 = 117649 \bmod 11 = 4$
5. Bob将B发给Alice
   Bob将4发给Alice

算法示例

6. Alice计算秘密密钥K1如下（知n=11,g=7,A=2,B=4,x=3; 但不知y=6）：
   $K1=B^x \bmod n$
   有 $K1 = 4^3 \bmod 11 = 64 \bmod 11 = 9$
7. Bob计算秘密密钥K2如下（知n=11,g=7,A=2,B=4,y=6; 但不知x=3 ）：
   $K2=A^y \bmod n$
   有 $K2 = 2^6 \bmod 11 = 64 \bmod 11 = 9$

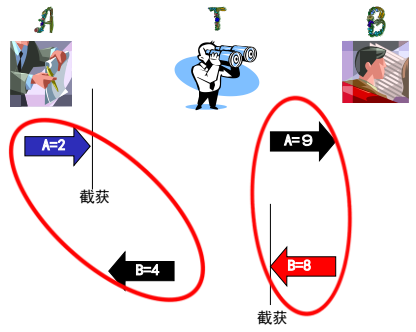
数学理论

- 安全性在于有限域中的**离散对数计算的困难性**

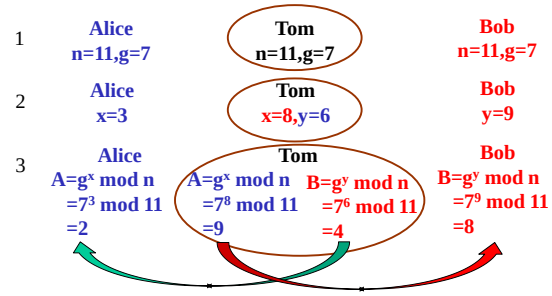
Alice在第6步计算	Bob在第7步计算
$K1=B^x \bmod n$ $B=g^y \bmod n$	$K2=A^y \bmod n$ $A=g^x \bmod n$
$K1=(g^y)^x \bmod n=g^{yx} \bmod n$	$K2=(g^x)^y \bmod n=g^{xy} \bmod n$
$K1=K2=K$	

- Alice与Bob交换n、g 、A 、B。根据这些值并不容易求出x（Alice知道x）和 y（ Bob知道y）,数学上对于**足够大的数**，求x与y是相当复杂的
- a ≡ b(mod n) a>0而0<b<n, b是a/n的余数，a为同余．例: 23≡11(mod 12)
- 求模指数$y=a^x \bmod n$ ---------- **容易**
- 求一个数的离散对数．求x，使$a^x \equiv b \pmod n$ ---- **很难**

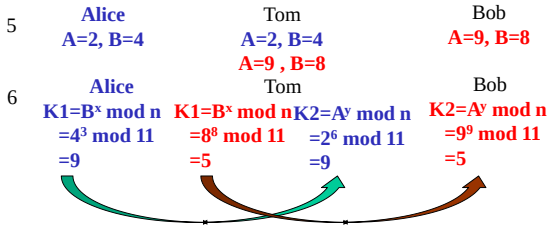
中间人攻击



例：中间人攻击



例：中间人攻击 （续）



- 如何克服？
  - 公钥证书
  - 数字签名

对称密钥的问题

- 多方通信（N方通信）
  - A、B (A-B) 1
  - A、B、C (A-B 、 A-C 、 B-C) 3
  - A、B、C 、D (A-B 、 A-C 、 A-D 、 B-C 、 B-D 、 C-D) 6
  - n **n*(n-1)/2**
  - 例： N=1000 ，所需密钥数量： 499500
- DES算法所需密钥数量为O(n^2)
- 密钥管理复杂，脆弱性
- 不支持身份认证

公钥密码体制：基本概念

- 非对称式（asymmetric）加密方法
  - 用两个不同的密钥来对信息进行加密和解密
  - public-key/two-key/asymmetric cryptography
- 两个密钥
  - 公开密钥（a public-key）：公开的，可用于加密消息，验证签名
  - 私有密钥（a related private-key）：秘密的，用于解密消息, 数字签名
- 很难从公开密钥得到私有密钥
- 非对称：用户加密消息或验证签名的密钥无法解密消息或进行签名。

公钥算法基本思想和要求

- 涉及到各方：**发送方、接收方、攻击者**
- 涉及到数据：**公钥、私钥、明文、密文**
- 公钥算法的条件：
  - **加密是可行的**
    - **产生一对密钥**是**计算可行**的
    - 已知**公钥和明文，产生密文**是**计算可行**的
    - 接收方**利用私钥来解密密文**是**计算可行**的
  - **破密是不可行的**
    - **利用公钥来推断私钥**是**计算不可行**的
    - 已知**公钥和密文，恢复明文**是**计算不可行**的
  - 加密和解密的顺序可交换**(可选)**

问题?

- 为什么需要公钥加密（Public-Key Cryptography）?
  - 密钥分发（key distribution）：共享对称密钥
  - 数字签名（digital signatures）：商业和个人应用
- 几个问题
  - 公钥密码是否比传统密码更抗密码分析?
    - 安全性取决于密钥长度；破解密码所需的计算量
  - 公钥密码是否能取代传统密码?
    - 公钥密码技术开销大
  - 公钥密码体制中的密钥分配是否更简单?
    - 中心代理，分发协议

非对称密钥（公钥）

- 多方通信（N方通信）
  - A、B(A-B) 2个私钥
  - A、B、C (A-B 、 A-C 、 B-C) 3个私钥
  - A、B、C、D (A-B 、 A-C 、 A-D 、 B-C 、 B-D 、 C-D) 4个私钥
  - n n
  - N=1000 1000个私钥
- 实际
  - 1000个锁，1000个公钥，1000个私钥
- 问题
  - 密钥难以做到一次一密；
  - 分组长度太大，使运算代价很高，尤其是速度较慢。

对称与非对称密钥加密比较

特征	对称密钥加密	非对称密钥加密
加密/解密使用的密钥	相同	不相同
加密/解密的速度	快	慢
得到的密文长度	等于或小于明文长度	大于明文长度
密钥协定与密钥交换	大问题	没问题
所需密钥数量和消息交换参与者个数的关系	大约为参与者个数的平方,伸缩性不好	等于参与者个数,伸缩性好
用法	主要用于加密/解密(保密性),不能用于数字签名(完整性和不可抵赖检查)	可以用于加密/解密(保密性)和数字签名(完整性和不可抵赖检查)

算法分类

- 大整数素因子分解系统（代表性的算法是RSA），应用最广泛的公钥系统
- 椭圆曲线加密算法（Elliptic Curve Cryptography，ECC）是基于离散对数的计算困难性。
- 数字签名算法（Data Signature Algorithm，DSA）是基于离散对数问题的数字签名标准，它仅提供数字签名功能，不提供数据加密功能。

RSA算法

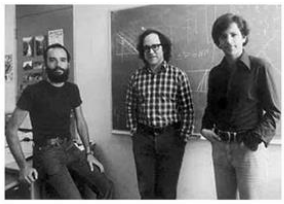
- 1977年由**R**onald Rivest、Adi **S**hamir和Len **A**dleman发明，1978年公布
- 是一种**块加密算法（分组密码）**
  - 明文和密文均是0~n-1之间的整数
- **应用最广泛**的公钥密码算法
- 只在美国申请专利，且已于2000年9月到期

RSA

- Ronald Rivest（罗纳德·李维斯特）和Adi Shamir（阿迪·萨莫尔）和Leonard Adleman（伦纳德·阿德曼）1977年一起发明了RSA公钥算法
  - Rivest：麻省理工(MIT)电子工程和计算机科学系教授
  - Shamir：以色列Weizmann科学学院应用数学系的教授
  - Adleman：南加州大学计算机科学以及分子生物学教授
- 美国计算机协会(ACM)将2002年图灵奖授予Ronald L. Rivest，Adi Shamir 和Leonard M. Adleman，以表彰他们在公钥加密算法上所做贡献。

"For their ingenious contribution for making public-key cryptography useful in practice."

2002年图灵奖：“RSA”




Ronald Rivest（罗纳德·李维斯特）
Adi **S**hamir（阿迪·萨莫尔）
Leonard **A**dleman（伦纳德·阿德曼）

RSA简介

- ◆素数 - - 只能被1和本身整除的数（3、5、7、11、13、17）
- ◆RSA算法思想：
  - ❖两个大素数**很容易相乘**
  - ❖而对得到的**积求因子则很难**
  - ❖RSA中的私钥和公钥基于**大素数（100位以上）**
  - ❖难度在于RSA选择和生成私钥与公钥

RSA密钥生成与使用

- ◆产生密钥对
  - ❖选择两个大素数p,q, p≠q
  - ❖计算n=pq,欧拉函数$\phi(n)=(p-1)(q-1)$
  - ❖**选择整数e（公钥，即加密密钥）,使gcd(e,ϕ(n))=1**
  - ❖**选择整数d（私钥，即解密密钥）,使d*e mod ϕ(n) =1**
- ◆公钥: KU={e,n}，私钥: KR={d,n}
- ◆使用
  - ❖加密: C = M^e mod n　　明文
  - ❖解密: M = C^d mod n　　密文

例：RSA密钥生成与使用

- ◆产生密钥对
  - ❖选择两个大素数p=7,q=17, p≠q
  - ❖计算n=pq=7*17=119,ϕ(n)=(p-1)(q-1)=6*16=96,
  96的因子有2,2,2,2,2和3，**因此e不能有2和3的因子**
  - ❖选择整数**e=5**（公钥，即加密密钥）,使gcd(e,ϕ(n))=1
  - ❖选择整数**d=77**（私钥，即解密密钥），
  使d*e mod ϕ(n) =1

  (5*77) mod 96 = 385 mod 96 =1
- ◆公钥: KU={e,n}=**{5,119}**，私钥: KR={d,n}=**{77,119}**
- ◆使用
  - ❖ **加密: $C = M^e \bmod n$**
  - ❖ **解密: $M = C^d \bmod n$**

例：RSA密钥生成与使用

使用公钥的加密算法

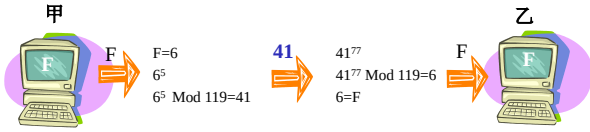
加密: $C = M^e \bmod n$

1. 用A=1、F=6等编码原字符
2. 公钥: KU={e,n}={5,119}
3. 将结果除以119，得到余数，即为密文

使用私钥的解密算法

解密: $M = C^d \bmod n$

1. 私钥: KR={d,n}={77,119}
2. 将结果除以119，得到余数，即为明文
3. 用A=1、B=2等译码原字符



RSA算法

- 块大小为k, $2^k < n \le 2^{k+1}$
  - 明文信息$M < n$
- 加密: $C = M^e \bmod n$
- 解密: $M = C^d \bmod n = M^{ed} \bmod n$
- 公钥: KU={e,n}, 私钥: KR={d,n}
- 上述算法需要满足以下条件:
  - 能够找到**e,d,n,**使得$M^{ed} \bmod n = M$, 对所有M<n
  - 计算M^e和C^d相对容易
  - **从e和n得到d是在计算上不可行的**

RSA密钥产生

- RSA实现
  - **素数的产生**
  - 大整数模运算
- 产生两个素数
  - 由于$n = pq$是公开的，所以，为了防止攻击者利用 n获得 p和 q，必须选择足够大的素数p和q
  - 大素数产生算法
- 选择e或者d，然后求出另一个

RSA安全性分析

- 结论
  - 已知的方法至少跟因子分解一样难度
  - 尚未发现多项式时间的因子分解算法
  - 因子分解的算法已经取得了长足进步
- 措施
  - 选择足够大的**n （1024或2048位以上）**
  - 并且使得 e, d之间相差不太大, 也不太小

椭圆曲线密码体制

- 1985年，N.Koblitz及V.S.Miller分别提出了椭圆曲线密码体制（ECC）
  - 解决椭圆曲线离散对数问题，对应有限域上椭圆曲线的群
- 与其他公钥密码体制相比，椭圆曲线密码体制的优点主要表现在以下4个方面
  - 密钥尺度较小
  - 参数选择比较灵活
  - 具有由数学难题保证的安全性
  - 实现速度较快
- 为满足电子认证服务系统等应用需求，国家密码管理局于2010年12月17日发布了SM2椭圆曲线公钥密码算法

国密算法

- 国密即国家密码局认定的国产密码算法，主要有SM1，SM2，SM3，SM4，密钥长度和分组长度均为128位
- SM1 为对称加密。其加密强度与AES相当。该算法不公开，调用该算法时，需要通过加密芯片的接口进行调用。
- SM2为基于ECC的非对称加密。该算法已公开。其签名速度与秘钥生成速度都快于RSA。SM2采用的是ECC 256位的一种
- SM3 消息摘要。类似MD5。该算法已公开。校验结果为256位。
- SM4 无线局域网标准的分组数据算法。对称加密，密钥长度和分组长度均为128位。

椭圆密码体制（ECC）的安全性

- 对椭圆曲线研究的时间短
- 椭圆曲线要求密钥长度短，速度快
- Certicom公司对ECC和RSA进行对比后发现，在实现相同安全性的情况下，ECC所需要的密钥量要比RSA少的多

ECC的密钥长度	RSA的密钥长度	Mips年
160	1024	1012
320	5120	1036
600	21000	1078
1200	120000	10168

ECC的应用

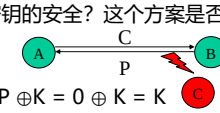
- Tor项目用它来帮助确保匿名
- 提供苹果iMessage服务的签名
- 加密DNS信息DNSCurve
- SSL/TLS协议验证安全网页浏览
- 使用ECC来提供对网络隐私必不可少的完全转发保密（perfect forward secrecy）
- Chrome或Firefox浏览器里访问HTTPS版本

作业

1. 攻击密码的两个通用方法是什么?
2. 分组密码和流密码的区别是什么?
3. 什么是数字签名?
4. 如何利用公钥体制进行秘密密钥的分发?
5. 私有密钥和秘密密钥有什么区别?

作业

- 阅读RFC4949, 理解网络安全基本概念
- 加密算法的两个基本要素是什么?
- 问题：某人用自己的密钥加密一个随机比特串（与密钥长度相同），采用异或运算，并通过通道发送结果。接收方得到密文后，和自己手里的密钥进行异或运算，并发回。如果某人接收到的这个是原始比特串，则证实两人拥有同一密钥。这种方法没有在信道上传递过密钥，是否能保证密钥的安全？这个方案是否有缺陷?



P⊕ K =C

C ⊕ P =P ⊕K ⊕P = P ⊕P ⊕K = 0 ⊕ K = K